Anonymizing Tables for Privacy Protection

Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rajeev Motwani,
Rina Panigrahy, Dilys Thomas, An Zhu
Stanford University*

We consider the problem of releasing tables containing personal records while ensuring individual privacy and data integrity. One of the techniques proposed in the literature is *k-anonymization*. A release is considered *k-anonymous* if the information for each person contained in the release cannot be distinguished from at least $k-1$ other persons whose information also appears in the release. We show that the problem of k-anonymization is NP-hard even when the attribute values are ternary. On the positive side, we give an $O(k)$-approximation algorithm for the problem. This improves upon the previous best known $O(k \log k)$-approximation [MW04]. In addition, we give a 1.5-approximation algorithm for the special case of 2-anonymity, and a 2-approximation algorithm for 3-anonymity.

The information age has witnessed a huge growth in the amount of personal data that can be collected and analyzed. This has led to an increasing use of data mining tools with the basic goal of inferring trends in order to predict the future. However, this goal conflicts with the desire for privacy of personal data. In many scenarios, access to large amounts of *personal data* is essential in order for accurate inferences to be drawn. For example, hospitals might wish to collaborate in order to catch the outbreak of epidemics in its early stages. This requires them to allow access to medical records of their patients. In such cases, one would like to provide data in a manner that enables one to draw inferences without violating the privacy of individual records.

One approach is to suppress some of the sensitive data values. This ensures complete data integrity, i.e., inferences can be made with 100% confidence (as compared to perturbation techniques). We study the k-anonymity model which was proposed by Samarati and Sweeney [Swe02, SS98]. Consider a database with n rows and m columns in which each entry comes from a finite alphabet Σ. For example, in a medical database, the rows represent individuals and the columns correspond to the different attributes. We would like to suppress some of the entries so that each row becomes identical to at least $k-1$ other rows. A suppressed entry is denoted by the symbol $*$. Since suppression results in the release of *less information and hence less utility*, we would like to *suppress as few entries as possible*.

We can view the database as consisting of n m-dimensional vectors: $x_1, x_2, \ldots, x_n \in \Sigma^m$. A *k-anonymous suppression* function t maps each x_i to $\tilde{x}_i$ by replacing some components of x_i by $*$, so that every $\tilde{x}_i$ is identical to at least $k-1$ other $\tilde{x}_j$'s. This results in a partition of the n row vectors into *clusters* of size at least k each. The cost of the suppression, $c(t)$ is the total number of $*$'s in all the $\tilde{x}_i$'s.

k-ANONYMITY: *Given $x_1, x_2, \ldots, x_n \in \Sigma^m$, obtain a suppression function t so that $c(t)$ is minimized.*

By reduction from EDGE PARTITION INTO TRIANGLES, we show that k-ANONYMITY is NP-hard even when the alphabet size $|\Sigma| = 3$. This improves upon the NP-hardness result of [MW04] for an alphabet size of n.

*Contact: kngk@cs.stanford.edu. Supported in part by NSF Grant ITR-0331640 and NSF Grant EIA-0137761.

On the positive side, we provide a $3k-3$-approximation algorithm for arbitrary k and arbitrary alphabet size. Given an instance of the k-anonymity problem, we create an edge-weighted complete graph, with vertices corresponding to rows in the database and Hamming distance between two rows as the edge weight. The algorithm creates a special forest on these vertices, with at least k vertices in each tree and of cost at most that of the optimal solution (OPT). Then some edges are deleted so that each component in the forest has between k and $3k-3$ vertices. Clearly the forest describes a feasible partition for the k-anonymity solution. In this solution, the number of $*$'s in each vertex is at most the cost of the component containing this vertex, since any attribute along which a pair of vertices differs appears on the path between the two vertices. Hence the total cost of the k-anonymity solution is within a factor of $3k-3$ from the forest cost and consequently OPT. More details are provided in the extended abstract [AFK+04].

For binary alphabet, we provide a polynomial time 1.5-approximation algorithm for 2-anonymity and 2-approximation algorithm for 3-anonymity, using polynomial time algorithms for obtaining minimum weight $[1,2]$-factor and 2-factor of a graph respectively.

References

[AFK+04] Aggarwal, Feder, Kenthapadi, Motwani, Panigrahy, Thomas, and Zhu. k-anonymity: Algorithms and hardness. Technical report, Stanford University, `http://dbpubs.stanford.edu/pub/2004-24`, April 2004.

[MW04] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of the ACM Symp. on Principles of Database Systems*, June 2004.

[SS98] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proc. of the ACM Symp. on Principles of Database Systems*, page 188, 1998.

[Swe02] L. Sweeney. k-Anonymity: A model for protecting privacy. In *International Journal on Uncertainty Fuzziness Knowledge-based Systems*, June 2002.